Journal of Theoretical Biology 279 (2011) 83–89

Contents lists available at ScienceDirect

Journal of Theoretical Biology

journal homepage: www.elsevier.com/locate/yjtbi

A novel clustering method via nucleotide-based Fourier power spectrum analysis

Bo Zhao [a], Victor Duan [b], Stephen S.-T. Yau [c,*]

[a] Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, Chicago, IL 60607, USA
[b] Illinois Mathematics and Science Academy, Aurora, IL 60506, USA
[c] Department of mathematical sciences, Tsinghua University, Beijing, 100084, P.R. China

ARTICLE INFO

Article history:
Received 12 October 2010
Received in revised form
2 March 2011
Accepted 22 March 2011
Available online 2 April 2011

Keywords:
Clustering
Fourier spectral method
Phylogenetic trees
DNA sequences
Moment vectors

ABSTRACT

A novel clustering method is proposed to classify genes or genomes. This method uses a natural representation of genomic data by binary indicator sequences of each nucleotide (adenine (A), cytosine (C), guanine (G), and thymine (T)). Afterwards, the discrete Fourier transform is applied to these indicator sequences to calculate spectra of the nucleotides. Mathematical moments are calculated for each of these spectra to create a multidimensional vector in a Euclidean space for each gene or genome sequence. Thus, each gene or genome sequence is realized as a geometric point in the Euclidean space. Finally, pairwise Euclidean distances between these points (i.e. genome sequences) are calculated to cluster the gene or genome sequences. This method is applied to three sets of data. The first is 34 strains of coronavirus genomic data, the second is 118 of the known strains of Human rhinovirus (HRV), and the third is 30 bacteria genomes. The distance matrices are computed based on the three sets, showing the distances from each point to the others. We used the complete linkage clustering algorithm to build phylogenetic trees to indicate how the distances among these sequence correspond to the evolutionary relationship among these sequences. This genome representation provides a powerful and efficient method to classify genomes and is much faster than the widely acknowledged multiple sequence alignment method.

1. Inspiration and motivation

Recently, there has been much research regarding methods to classify genomes into correct biological groups. A prominent method in use today is the multiple sequence alignment method. However, while this method is widely recognized as an accurate means of grouping genomes, it is extremely time consuming and can take up to several days or more depending on the amount of data being examined. In this project, we seek to find an alternative method to cluster genomes that does not require vast computational power.

Multiple sequence alignment arranges the genomes and finds the differences and similarities in the nucleotide data. Typically, parts of the genome that are compared include DNA and RNA. This allows for accurate determination of functional, structural, or evolutionary relationships between genomes.

Our method is based on Fourier analysis. Fourier analysis has been used in previous research. While researching applications of Fourier analysis, we see that it has been used before to differentiate exons from introns using 3-base periodicity (Yin and Yau, 2005, Prediction of protein coding regions by the 3-base periodicity analysis of a DNA sequence, 2007). Yin and Yau (2007) showed that the discrete Fourier transform has powerful uses in extracting useful information from genomic data. This leads us to wonder if we could use the Fourier power spectrum to introduce a new method to cluster genomes rather than use multiple sequence alignment.

To do this, we propose using a method involving the discrete Fourier transform and moment vectors. A quicker, but still accurate method to cluster genomes would allow people to better understand evolution of organisms as well as the relationships between various genomes because multiple sequence alignment takes too much time to be a reasonable approach for a huge data set.

Our research provides an efficient method to cluster genomes with much less time-consumption compared to the multiple sequence alignment method. This work can lead to discoveries in the world of biology as scientists are able to more quickly analyze the relationships between various organisms.

2. Methods

2.1. Reinterpretation of genomic data

We begin with genomic data composed of the nucleotides adenine (A), cytosine (C), guanine (G), and thymine (T). These data

* Corresponding author.
E-mail address: yau@uic.edu (S.S. Yau).

0022-5193/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jtbi.2011.03.029

are easier to work with if they are in numeric sequences. As a result, we convert the genomic data of length N into four separate binary indicator sequences—one for each nucleotide. Each value will be either 1 or 0; 1 if that nucleotide appears in that position and 0 otherwise.

We define the indicator sequence $u_\alpha(n)$ of the DNA sequence:

$$u_\alpha(n) = \begin{cases} 1, & \alpha \text{ appears at location } n\text{of the DNA sequence,} \\ 0, & \text{otherwise,} \end{cases}$$

where $\alpha \in I = \{A,T,C,G\}$, $n = 0,1,\ldots,N-1$, and N is the length of the DNA sequence (i.e., number of base pairs).

For example, for the sequence ACTGTCGATT, the corresponding indicator sequences are

u_A : 1000000100
u_C : 0100010000
u_G : 0001001000
u_T : 0010100011.

2.2. Discrete Fourier transforms

After the genomic data have been converted into these indicator sequences, they can be manipulated with mathematical methods. The discrete Fourier transform is applied to each indicator sequence $f(n)$ and a new sequence of complex numbers, called $F(k)$, is obtained:

$$F(k) = \sum_{n=0}^{N-1} f(n) e^{-i(2\pi/N)kn}, \quad \text{for} \quad k = 0,1,2,\ldots,N-1, \tag{1}$$

where $f(n)$ is any of the four indicator sequences, u_A,u_C,u_G,u_T, introduced in the last subsection. The discrete Fourier transform thus converts a sequence in state space into a sequence in frequency space (Peebles, 2000), better revealing hidden statistical characteristics of the original data.

2.3. Power spectrum analysis

It is easier to work with the power spectrum of the sequence, rather than the original discrete Fourier transform. The power spectrum (PS) for frequencies $k = 0,1,2,\ldots,N-1$ is defined as

$$PS(k) = |F(k)|^2. \tag{2}$$

Although this loses some of the information from the complex numbers such as the angle/direction, the power spectrum still contains significant information while being simpler to analyze.

2.4. Moment vectors

It is a challenging problem to compare various genomes by only looking at the power spectra data. As a result, we must look at the mathematical moments of the data involved. There are several ways to look at moments as well as several ways to normalize them. The usual moments are $\tilde{M}_j = (1/N)\sum_{k=0}^{N-1}(PS(k))^j$ for $j = 1,2,\ldots$.

However, $\tilde{M}_j$ increases very rapidly as j increases. Consequently, when comparing genomes based on multiple moments, the higher moments hold much greater weight due to their magnitude. To compensate for this, we use various normalization factors. In the above $\tilde{M}_j$ definition, N is the denominator, but it can be altered to vary with the order of the moment as well. This can be done as follows:

$$M_j = \frac{1}{N^j} \sum_{k=0}^{N-1} (PS(k))^j, \quad j = 1,2,\ldots. \tag{3}$$

Additionally, N is the length of the whole sequence, but we split the sequence into four separate vectors for the four nucleotides. As a result, each nucleotide has its own associated sequence length, or the number of that specific nucleotide. These values will be known as N_A,N_C,N_G, and N_T. Because the whole genome consists of nucleotides A, C, G, and T, they can also be incorporated into the equations to normalize the moments to make higher moments at similar values to the lower moments. This is important at later steps when we are comparing the genomes. The new lengths can be incorporated into the normalization in the following equation:

$$M_j^A = \frac{1}{N_A^{j-1} N^{j-1}} \sum_{k=0}^{N-1} (PS(k))^j. \tag{4}$$

Moreover, instead of only analyzing $(PS(k))^j$, central moments can be calculated by instead analyzing the value of the difference between the power spectrum value at a point k and the mean of all values of the power spectrum. In other words, we would do the following to calculate central moments with a normalizing factor of just $(1/N)$:

$$\text{Mean} = \frac{1}{N} \sum_{k=0}^{N-1} PS(k),$$

$$\text{CM}_j = \frac{1}{N} \sum_{k=0}^{N-1} (PS(k) - \text{Mean})^j. \tag{5}$$

Just as we altered the normalizing factors with the regular moments, we can do the same for the central moments to get several other equations as shown below:

$$\text{CM}_j^A = \frac{1}{N_A^{j-1} N^{j-1}} \sum_{k=0}^{N-1} (PS(k) - \text{Mean})^j. \tag{6}$$

2.5. Genomic comparisons

These equations for calculating moments are used to analyze various genomic data. To do this, several moments of each genome are computed, which is done in C++, and we built the phylogenetic trees using the function hclust in R (R Development Core Team, 2008), where the complete linkage clustering algorithm is used. Specifically, we calculate the first few regular and central moments of the genome for each nucleotide and assign that genome a point in Euclidean space with those moments as the coordinates. Then, a distance matrix can be generated from the pairwise distances of the genomes using Euclidean distance. A distance matrix can then be used in clustering algorithms to separate genomes into various clusters or groups. Here, we have used the complete linkage clustering algorithm and the average linkage clustering algorithm. In complete linkage clustering, the distance between two clusters is computed as the distance between the farthest two elements in the two clusters (Dawyndt et al., 2005). On the other hand, the distance between two clusters in average linkage clustering is defined as the average pairwise distance between the points of two clusters. The clustering algorithms then create a phylogenetic tree to show how the genomes are grouped together.

We realize that just a few moments may not be enough to give stable, accurate clustering results. However, we do not need too many moments either. This is because as the number of moments increases, the clustering results will quickly stabilize due to the fact that the magnitude of the higher moments quickly drops close to 0, rendering their effect negligible. We take 20 moments here: the first three regular moments, and the second and third

central moments for each of four nucleotides. This gives each nucleotide five moments, resulting in a 20-dimensional point in Euclidean space. The same combination gives good results for all the examples in the present paper (coronavirus, human rhinovirus, and bacteria), although fewer moments may be needed for the (smaller) coronavirus genomes to achieve the same results.

All computations in this paper are done on a Dell laptop equipped with Intel i3 Processor under Windows 7 Home Premium with 4 GB RAM, together with the statistic computing software R (Version 2.9.2) and Microsoft Visual Studio 2008 (with C++).

3. Results and discussions

To verify that the data found using these methods really corresponds to true biological groups from literature, we apply our moment vectors and clustering algorithms first for coronavirus and human rhinovirus to compare with existing phylogenetic trees, and then for bacterial species to generate biologically correct phylogenetic trees.

3.1. Clustering method

We sought to verify some clustering results with our own method. To do this, we apply our method to three sets of genomes: coronaviruses, human rhinoviruses, and bacteria. To cluster various genomes, we experimented with various combinations of types of moment, normalization factor, and clustering methods. Accurate results came from a combination of the use of regular moments and the use of central moments of all four nucleotides. The normalizing factor used was $1/(N^{j-1}N_m^{j-1})$ with m being the nucleotide of that particular moment. The first three regular moments of each nucleotide were calculated with Eq. (4). Afterwards, the second and third central moments were calculated with Eq. (6). This gives each nucleotide five moments, resulting in a 20-dimensional point in Euclidean space. An Euclidean pairwise distance matrix is then calculated to use with clustering algorithms (Dawyndt et al., 2005) in the statistical program R.

To evaluate complete linkage clustering algorithm and the average linkage clustering algorithm, the cophenetic correlation coefficients are calculated. The values are shown in Table 1. As you can see, average linkage clustering has higher cophenetic coefficients than complete linkage clustering, but the difference is very small, so we construct the phylogenetic trees with both two algorithms.

3.2. Coronavirus (respiratory disease)

We begin by studying the taxonomy of coronavirus and how the complete genomes of 30 separate coronaviruses cluster into groups. Just as Yu et al. (2010) suggested, we also included four non-coronavirus genomes to act as an outgroup separate from the groupings of the coronaviruses. The accession numbers, abbreviations, group numbers, and descriptions are shown in Table A in the Supplementary materials.

Table 1
Comparison of cophenetic coefficients.

	Human rhinovirus	Coronovirus	Bacteria
Average linkage clustering	0.93332	0.98939	0.86881
Complete linkage clustering	0.87216	0.98799	0.86481

In Yu et al. (2010), these 34 genomes were separated into five groups and an outgroup. With 34 points, we then calculated the Euclidean distance matrix, and used both the complete linkage clustering and average linkage clustering algorithms (Dawyndt et al., 2005) in the statistical program R. The results are shown in the phylogenetic trees shown in Fig. 1.

Traditional clustering has shown the majority of these groups to be correct. However, a few of the more newly discovered coronaviruses have been studied to decide which group they belong to. Yu et al. (2010) and van der Hoek et al. (2004) agree with the placing of human rhinovirus NL63 into Group 1. However, as Fig. 1 shows, HCoV-NL63 is slightly separated from the other two members of Group 1, but close enough to be considered part of the same group, which is consistent with past work.

Another newer coronavirus, human rhinovirus HKU1, has also been debated about recently. Woo et al. (2005) contended that it belonged to Group 2 as it had certain characteristics of Group 2 coronaviruses (Woo et al., 2005). However, Woo et al. (2005) also noted that the proteins of HCov-HKU1 coronavirus are not very closely related to those of other Group 2 coronaviruses (Woo et al., 2005). As a result, HCoV-HKU1 is identified as a distinct part of the group of coronaviruses, leading Yu et al. (2010) to place it in a separate group between the SARS group (Group 4) and Group 2. As shown in Fig. 1, our method also shows HCoV-HKU1 in a separate group. In the phylogenetic tree in Fig. 1, Group 5 is close to both Group 4 and Group 2 but still distinguishable from both groups, so we agree that Group 5 should be separate from Group 2.

All in all, the clustering of these 30 coronavirus genomes is accurate according to previous clustering results.

3.3. Human rhinovirus (common cold)

After confirming that our method accurately clustered the genomes of various coronaviruses, we have also tried our method on a large set of human rhinovirus (HRV) genomes. Past work has shown that these HRV genomes can be split into three clades: HRV-A, HRV-B, and HRV-C (Palmenberg et al., 2009). Palmenberg et al. (2009) clustered the genomes into these groups with the multiple sequence alignment method, which takes vast amounts of time and computational power (Palmenberg et al., 2009). We attempt to generate these same results with our method while utilizing much less time. We used the complete genomes of 116 HRV serotypes as well as the three outgroup genomes suggested by Palmenberg et al. (2009). The accession numbers and abbreviations of these genomes are shown in Table B in the Supplementary material.

Palmenberg et al. (2009) clustered these genomes into three groups and an outgroup. We can generate these results using our method via discrete Fourier transform. We used the same combinations of moment, normalization factor, and clustering methods as used for coronavirus and produced the phylogenetic trees shown in Fig. 2.

Traditional grouping has shown these groups to be correct (Palmenberg et al., 2009). Our method was still able to differentiate the groups of HRV into the three clades and the outgroup.

3.4. Bacterial species

Bacteria genome lengths are millions of base pairs long. Due to their complexity, bacteria genomes are a good test to see whether a clustering method can handle huge sequences. In fact, most methods cannot handle bacteria genomes. As a result, we will also apply our clustering algorithm to 30 bacterial genomes from eight families: Enterobacteriaceae, Staphylococcaceae, Rhodobacteriaceae, Burkholderiaceae, Bacilleceae, Spirochaetaceae, Clostridiaceae, and Desulfovibrionaceae. Each bacterial genome sequence used in this

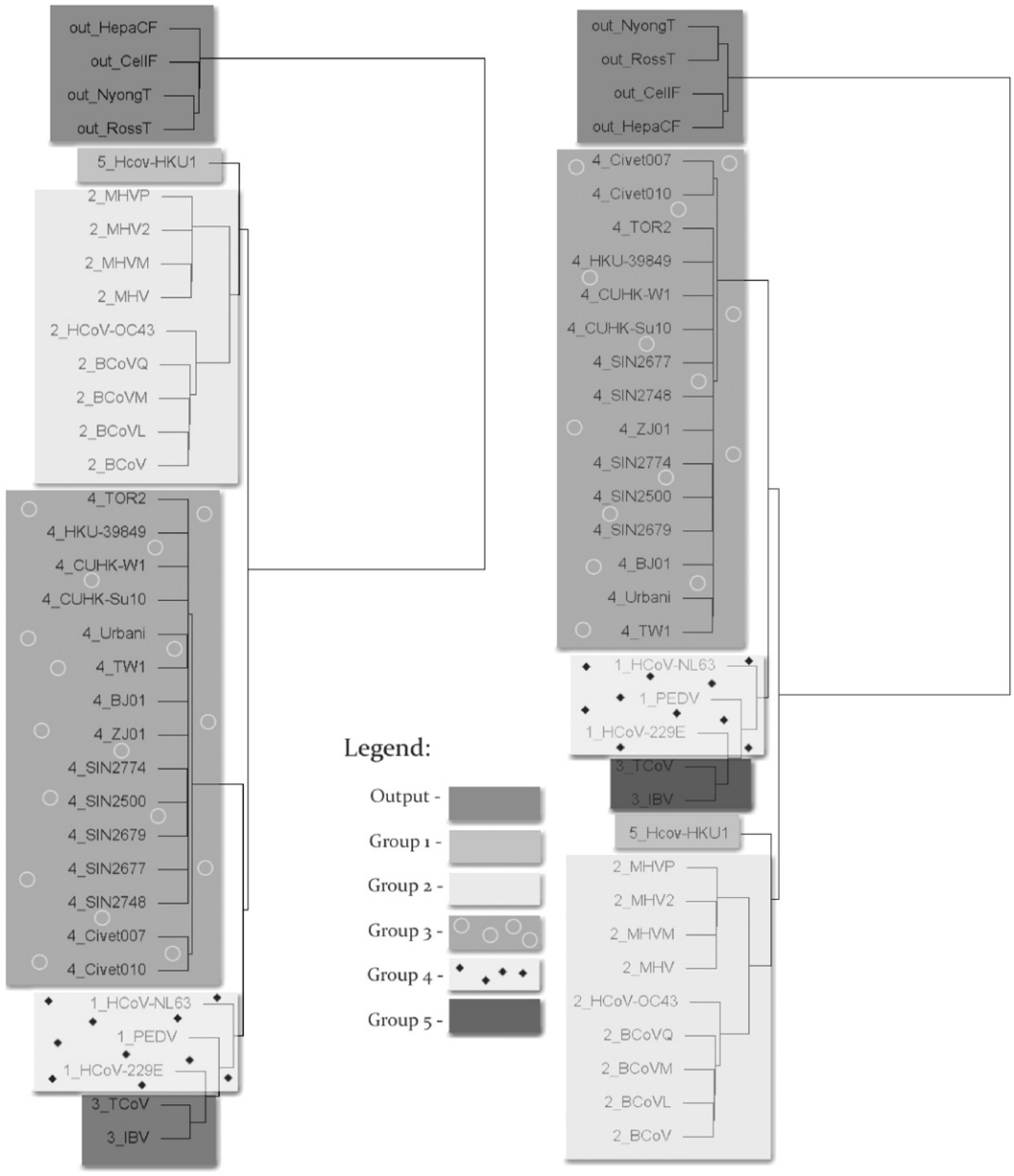

Fig. 1. These phylogenetic trees show the 30 coronavirus genomes as well as the four outgroup genomes. Our method has split them into the correct groups: outgroup and Groups 1-5.

present study (around the order of four million base pairs) is significantly longer than those of the coronavirus and Human rhinovirus genomes used in the previous two subsections. A list of the families, species, and accession numbers of these 30 bacteria is shown in Table C in the supplementary materials.

We cluster the genomes with our Fourier power spectrum analysis method. We use the same combinations of moment, normalization factor, and clustering methods as in the previous two examples. The phylogenetic tree is shown in Fig. 3.

As the tree shows, the bacteria can be separated into the correct families through the use of our Fourier power spectrum analysis method and our method is able to accurately group the genomes according to biological information. Furthermore, our method was capable of running on such large sequences in around five minutes. On the other hand, all traditional methods involving multiple sequence alignment require significantly more time and computational power, rendering it impossible for a personal computer to use multiple sequence alignment to cluster bacteria genomes.

3.5. Phylogenetic trees

Here, the phylogenetic trees generated from the three examples above are shown. In each figure, two trees are shown: one created by complete linkage clustering and one created by average linkage clustering. For the sake of clarity, the trees on the left will be created by complete linkage clustering and the trees on the right will be those created by average linkage clustering.

Fig. 1 shows the trees generated for the 30 coronavirus genomes and four outgroup genomes. Clearly, our method was able to cluster the genomes into the correct five groups. Furthermore, the trees created by complete linkage clustering and

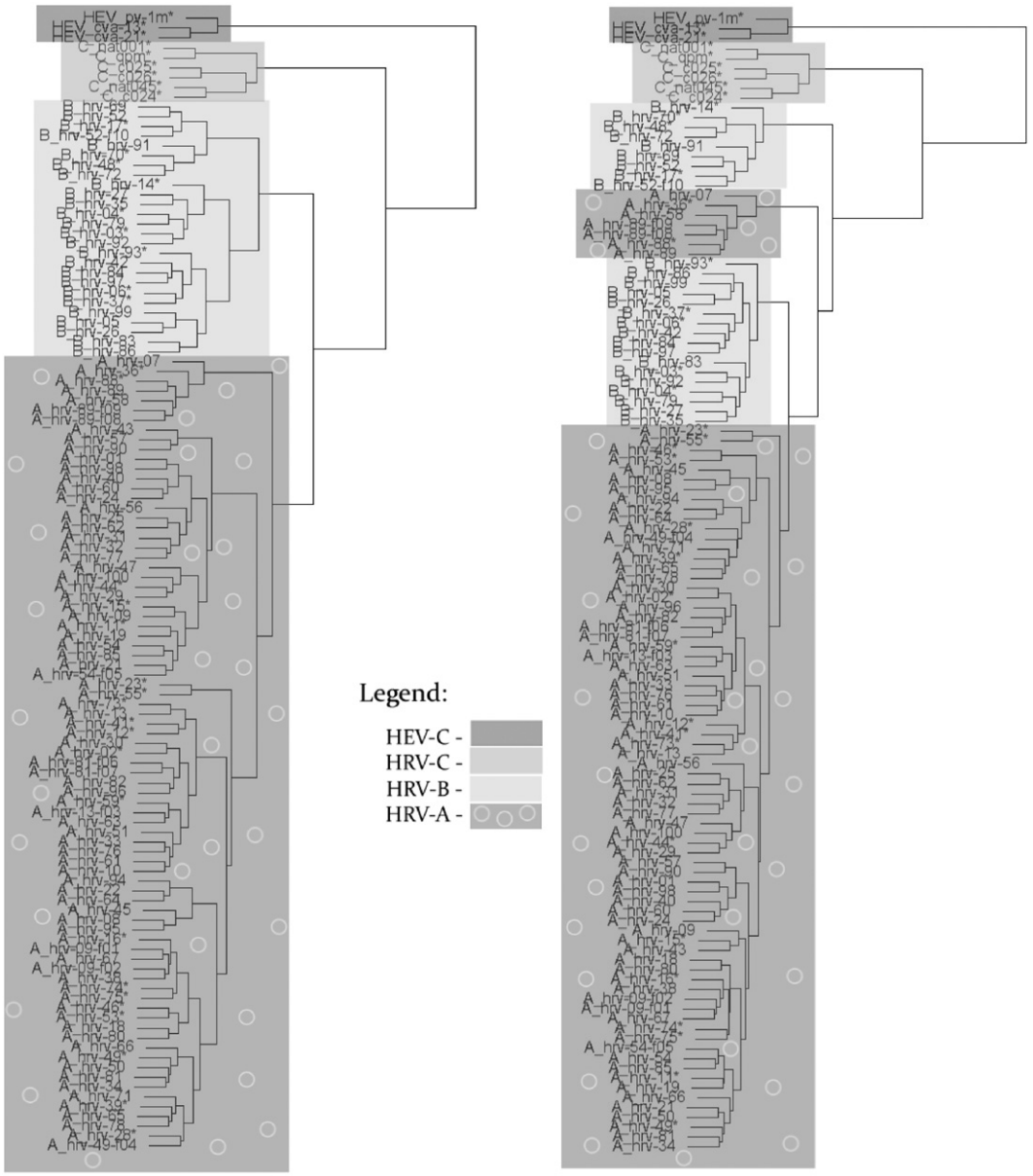

Fig. 2. These phylogenetic trees show the 116 human rhinovirus genomes as well as the three outgroup genomes. Our method has split them into the correct groups (HRV-A, HRV-B, HRV-C, and HEV-C (outgroup)) in complete linkage clustering, but there are seven misplaced HRV-A genomes in the tree created by average linkage clustering.

average linkage clustering are extremely similar. The genomes are correctly clustered, only the orientation of the tree is different.

Fig. 2 shows the trees generated for the 113 HRV genomes and three HEV (outgroup) genomes. Using complete linkage clustering, the genomes were correctly clustered into HRV-A, HRV-B, HRV-C, and HEV-C. However, the tree created using average linkage clustering shows subtle imperfections. A small group of HRV-A genomes is misplaced and put between the HRV-B genomes. Those genomes were originally on the edge of the HRV-A cluster in complete linkage clustering. When switching to average linkage clustering, the clusters changed. This mistake in the clustering tree shows that although average linkage clustering has a higher cophenetic coefficient, complete linkage clustering still provides better clustering results that match with previously proven results.

Finally, Fig. 3 shows the trees generated for 30 bacterial genomes from the families Enterobacteriaceae, Staphylococcaceae, Rhodobacteriaceae, Bacilleceae, Burkholderiaceae, Spirochaetaceae, Clostridiaceae, and Desulfovibrionaceae. As shown in the trees, both complete linkage clustering and average linkage clustering provide accurate clusterings. As with Fig. 1, the only difference is the orientation of the groups. In this case, the families of Enterobacteriaceae and Bacilleceae are always together. Meanwhile, Staphylococcaceae, Rhodobacteriaceae, Burkholderiaceae, Spirochaetaceae, Clostridiaceae, and Desulfovibrionaceae bacteria are always clustered together.

As can be seen from the phylogenetic trees generated by our method, both complete linkage clustering and average linkage clustering are able to provide accurate clusterings. However, with the human rhinovirus situation, average linkage clustering was unable to accurately group a few genomes while complete linkage clustering was able to. Although average linkage clustering provided a higher cophenetic coefficient than complete linkage clustering, complete linkage clustering provided superior results. This is because cophenetic analysis is not a definitive method to select the best clustering method. In the case of genome

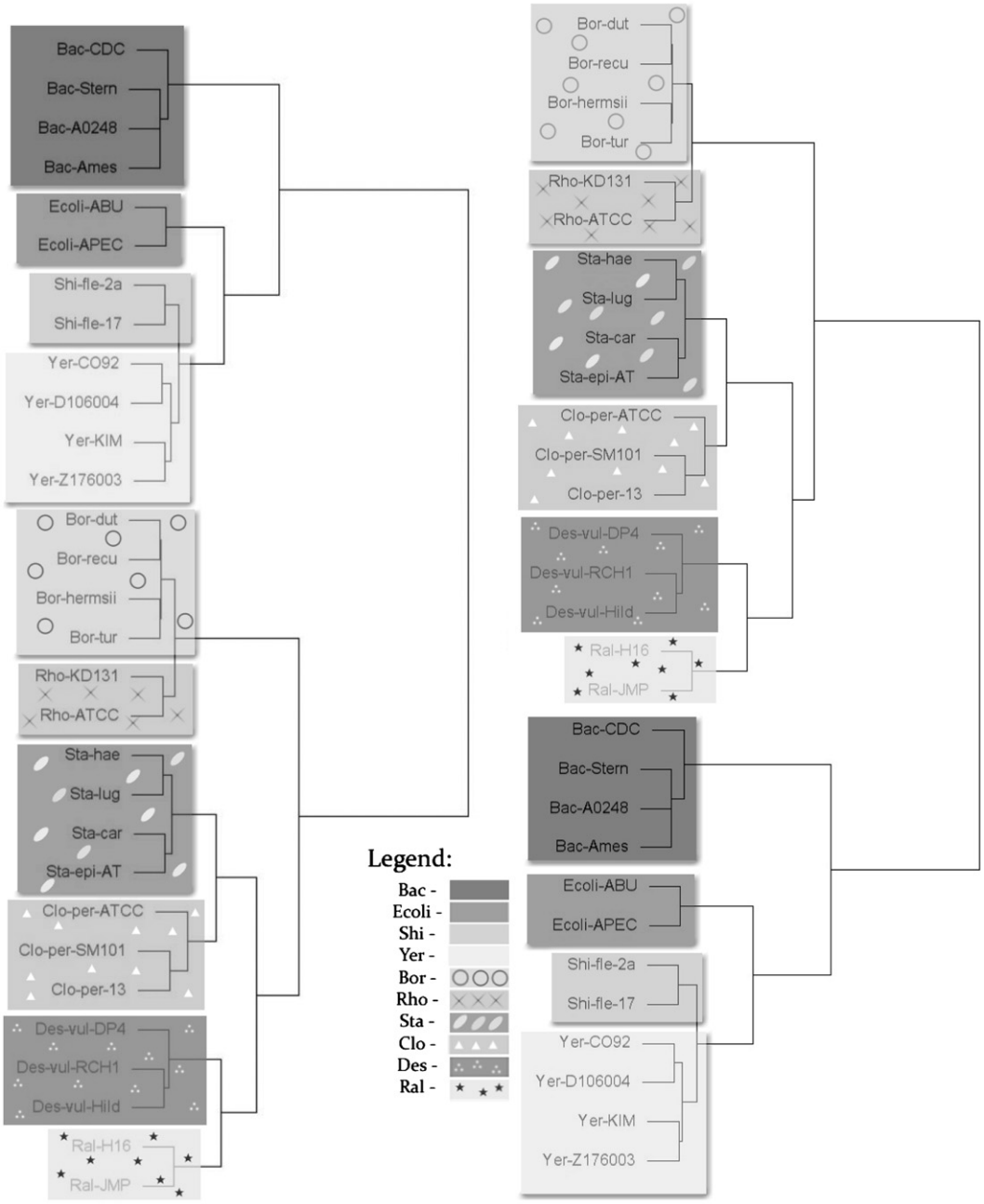

Fig. 3. These phylogenetic trees show the 30 bacteria genomes. Our method has split them into the correct families: Enterobacteriaceae, Staphylococcaceae, Rhodobacteriaceae, Burkholderiaceae, Bacilleceae, Spirochaetaceae, Clostridiaceae, and Desulfovibrionaceae.

clustering, we see that complete linkage clustering is able to provide better results in spite of its lower cophenetic coefficients. As a result, we propose that complete linkage clustering be used regardless of the cophenetic analysis results.

Overall, our method of clustering via Fourier power spectrum analysis provides an accurate way to cluster genomes.

3.6. Comparing speed to multiple sequence alignment

Now that this method of clustering via the discrete Fourier transform has been shown to be capable of calculating accurate results, its speed is now compared to the most prominent method of clustering: multiple sequence alignment. Programs used for multiple ClustalW (Brown et al., 2007), Muscle (Edgar, 2004), MAFFT (Katoh et al., 2009) and MISHIMA (Kryukov et al., 2010). Each program gives slightly different results and has its own strengths and weaknesses. CLUSTALW was among the first programs to produce results of multiple sequence alignment. Consequently, it is the most widely used method and gives very accurate results. However, it is very slow. MUSCLE is very similar to CLUSTALW and runs at a similar speed and gives similarly accurate results. MAFFT is a compromise between speed and accuracy. It runs much more quickly than CLUSTALW and MUSCLE, but sacrifices a small amount of accuracy to achieve these results. Mishima is a relatively newer program that also compromises speed and accuracy.

Our method defined in this paper is now compared to MAFFT and Mishima in terms of speed to show that our method is faster than the usual methods of clustering genomes. Our method is not compared to MUSCLE or CLUSTALW because those methods are

Table 2
Comparison of computing times (in s).

Computing time	MAFFT	MISHIMA	Our method
Human rhinovirus	255	1868	5
Coronovirus	1701.5	8994	12
Bacteria	N/A	N/A	298

significantly more time consuming and so should not be compared with our method as they attempt to achieve different goals. MUSCLE and CLUSTALW hope to achieve the best possible results with no regard to the speed or efficiency of their process. Meanwhile, MAFFT, Mishima, and our method hope to maximize accuracy while still maintaining speed and efficiency. As a result, our method deserves to be compared with Mishima and MAFFT. The comparison results are shown in Table 2.

When running to create a phylogenetic trees, comparisons were seen between our method, MAFFT, and Mishima. For the human rhinovirus genomes, MAFFT required 255 s, Mishima required 1868 s, and this paper's method required only 5 s. Furthermore, when running to create a phylogenetic tree for the coronavirus genomes, MAFFT required 1701.5 s, Mishima required 8994 s, and our method required a mere 12 s. Finally, MAFFT and Mishima were unable to run on the bacterial genomes on our personal computer due to the sheer size of the sequences. However, our method was able to complete the job in about 5 min (298 s).

As can be shown from the times required to generate phylogenetic trees, the method proposed in the present paper is significantly faster than MAFFT and Mishima.

4. Conclusions

After working with the discrete Fourier transform and various moment equations and clustering algorithms, we have arrived at a procedure that is able to quickly and accurately cluster various groups of genomes including coronaviruses, human rhinoviruses and bacteria into their correct biological groups. This method works by converting each DNA sequence into a point in a moment space and using the Euclidean distances between these points to provide a measure of closeness or relationship. Instead of taking hours as the multiple sequence alignment method requires, our method requires at most a few minutes to finish its calculations and to draw the phylogenetic tree depending on the number of sequences and the computer used. This gives significant advantages in doing research and experimenting.

Acknowledgment

Victor Duan would like to thank Shengqiang Xu for helpful discussions about C++ programming.

Appendix A. Supplementary data

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.jtbi.2011.03.029.

References

Brown, N.P., Larkin, M.A., Blackshields, G., 2007. Clustal w and clustal x version 2.0. Bioinformatics 23 (21), 2947–2948.
Dawyndt, P., De Meyer, H., De Baets, B., 2005. The complete linkage clustering algorithm revisited. Soft Computing 9 (May), 385–392.
Edgar, R.C., 2004. MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Research 32 (5), 1792–1797.
Katoh, K., Asimenos, G., Toh, H., 2009. Multiple alignment of dna sequences with MAFFT. Methods Molecular Biology 537, 39–64.
Kryukov, K., Saitou, N., 2010. MISHIMA – a new method for high speed multiple alignment of nucleotide sequences of bacterial genome scale data. BMC Bioinformatics 11 (142).
Palmenberg, A.C., Spiro, D., Kuzmickas, R., Wang, S., Djikeng, A., et al., 2009. Sequencing and analyses of all known human rhinovirus genomes reveal structure and evolution. Science 324, 55–59.
Peebles, P., 2000. Probability, Random Variables, and Random Signal Principles, fourth ed. McGraw-Hill.
R Development Core Team, 2008. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, ISBN: 3-900051-07-0.
van der Hoek, L., Pyrc, K., Jebbink, M.F., Vermeulen-Oost, W., Berkhout, R.J., Wolthers, K.C., et al., 2004. Identification of a new human coronavirus. Nature Medicine, pp. 368–373.
Woo, P.C., Lau, S.K., Chu, C.-m., Chan, K.-h., Tsoi, H.-w., Huang, Y., et al., 2005. Characterization and complete genome sequence of a novel Coronavirusa coronavirus HKU1a from patients with pneumonia. Journal of Virology 79, 884–895.
Yin, C., Yau, S.S.-T., 2005. A Fourier characteristic of coding sequences: origins and a non-fourier approximation. Journal of Computational Biology 12, 1153–1165.
Yin, C., Yau, S.S.-T., 2007. Prediction of protein coding regions by the 3-base periodicity analysis of a DNA sequence. Journal of Theoretical Biology, 687–694.
Yu, C., Liang, Q., Yin, C., He, R.L., Yau, S.S.-T., 2010. A novel construction of genome space with biological geometry. DNA Research, 1–14.